An Effective Strategy for Greedy Behavior in Wireless Ad hoc Networks

Soufiene Djahel[∓], Farid Naït-Abdesselam[∓] and Damla Turgut [±]

[∓]LIFL – UMR CNRS USTL 8022 – IRCICA
University of Lille , France
{soufiene.djahel, farid.nait-abdesselam }@lifl.fr

[±]School of Electrical Engineering and Computer Science
University of Central Florida, USA
turgut@eecs.ucf.edu

Abstract— **While the problem of greedy behavior at the MAC layer has been widely explored in the context of wireless local area networks, its study for multi-hop wireless networks still almost an unexplored and unexplained problem. Indeed, in a wireless local area network, an access point mostly forwards packets sent by wireless nodes over the wired link. In this case, a greedy node can easily get more bandwidth share and starve all other associated contending nodes by intelligently manipulating the MAC layer parameters. However, in wireless ad hoc environment, all packets are transmitted in a multi-hop fashion over wireless links. Therefore, if a greedy node behaves similarly as in WLAN case, trying to starve its neighbors, then its next hop forwarding node will also be prevented to forward its own traffic, which leads to an end-to-end throughput collapse.**

In this paper, we show that in order to have a more beneficial greedy behavior in wireless ad hoc networks, a node must adopt a different approach than in WLAN to achieve a better performance of its own flows. We then present a strategy to launch such greedy attack in a proactive routing based wireless ad hoc network. Through the extensive simulations, the obtained results show that by applying the proposed algorithm, a greedy node can gain more bandwidth than its neighbors and keep the end-to-end throughput of its own flows highly reasonable.

Keywords – **Greedy behavior, MAC layer misbehavior, Ad hoc networks, conflict graphs.**

I. Introduction

The increase in computation power, the compactness of size, incorporation of mobility and ease of connectivity from anywhere are amongst the major factors that resulted in tremendous growth of handheld devices in recent years. From cordless phones to cellular networks and from WiFi to sensors, the wireless medium has become the preferred backbone of today's deployed networks. The nodes in Mobile Ad hoc Networks (MANET) communicate directly over the wireless links if they are within the transmission range of each other. Otherwise, the communication is achieved by the intermediate nodes relaying the messages from source to destination pairs. The properties of MANET, such as shared wireless medium, open network architecture, stringent resource constraints and rapidly changing topology make this type of network vulnerable to various attacks at different layers, especially at MAC layer in which these attacks can be launched easily. Therefore, the task of securing such a network remains hard and requires careful investigation.

Since IEEE 802.11 MAC protocol, as described in [3], is commonly used by wireless nodes to access the medium, any misbehavior at this level may affect the proper functioning of the network. The serious damage caused by MAC layer misbehavior has received considerable research attention leading to an in depth investigation and analysis of its root causes, such as the works done by Bellardo and Savagein [7] and Gupta et al. [4]. As a result of this investigation, some pioneering contributions have been proposed in the literature to cope with this problem such as Cardenas et al.[11] , Kyasanur and Vaidya [12] and Raya et al. [14]. These earlier works have identified several types of MAC layer misbehavior and proposed countermeasures to detect or prevent such misuse. However, their solutions are based on the assumption that the greedy node behaves similarly in MANET as in WLAN. This assumption is neither realistic nor sustainable. Moreover, it may even disrupt the performance of its own traffic as we will show in this paper. Therefore, the existing protocols do not offer fully satisfactory solutions in responding to the concern of greedy behaviors in MANET.

The majority of the previous protocols cited above are based on monitoring the behavior of one hop neighbors to detect any misuse of the protocol rules. The monitoring task is carried by a trustworthy entity such as in DOMINO by Raya et al. [14], which necessitates an election process to select this entity among the MANET nodes. Others such as Kyasanur and Vaidya [12] and Guang et al. [15] have devised new algorithms to schedule the access to the medium–they make the backoff value to be used by the monitored node easily predictable or known in advance. Applying our greedy strategy makes these schemes less effective since the collisions provoked by the greedy node with its neighbors disturb the monitoring operation, hence it hides its violation of the protocol rules and keep gaining more bandwidth at the expense of the well behaved nodes.

The remainder of the paper is organized as follows. Section II provides a comparison between greedy behavior in WLAN and MANET. Our proposed greedy strategy in MANET is presented in Section III. In section IV, we present and discuss the obtained simulation results. Finally, we conclude in Section V.

II. Greedy behavior impact on network performance: WLAN versus MANET

In this section we emphasize the major difference between the greedy behavior in WLAN and MANET. In other words, we try to answer the following question: Are the damages induced by greedy nodes in WLAN and MANET similar?

As illustrated in Fig. 1, the destination of a flow in WLAN can be either a far away node or the one attached to the same access point (AP). In the former case, the source node of the flow f_1 tries to gain the entire bandwidth regardless of the decrease in its neighbor's throughput. This is due to the fact that its next hop (AP) forwards the packets of the flow f_1 through a wired link, independent from the wireless ones (no transmission conflict exist between those links). The flow f_2 is similar to the case of the flow f_3 in MANET, any attempt of the flow's source node A or an intermediate node B to dominate the medium deprives its next hop from forwarding the received packets. Consequently, the flow's performance collapses sharply. Furthermore, the impact of this misbehavior may propagate to affect other flows crossing through the nodes in contention with the greedy node.

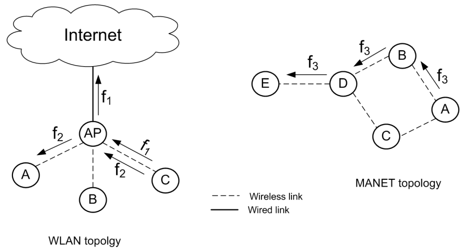


Figure 1: Greedy behavior: WLAN vs. MANET.

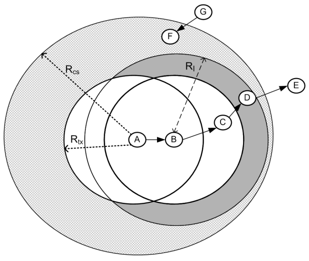


Figure 2: Propagation of greedy behavior's impact in MANET.

To illustrate this phenomenon related to radio wave propagation, let us consider the network topology given in Fig. 2. In this figure, R_{tx} and R_{cs} represent the transmission and carrier sensing ranges of node A, respectively. The lightly shaded area represents the region which is not covered by RTS/CTS handshake between A and B. Note that any transmission initiated from a node within this region may not interfere with packet reception at node B as these nodes are out of its interference area, represented by the darker region which is delimited by the interference range R_I. Despite that, the nodes within the lightly shaded area have to differ their transmissions since they sense the medium busy due to node A's transmission. As a result, if the sender node A misbehaves and monopolizes the medium for a long duration, all the transmissions over the links where at least one node is within the lightly shaded area are delayed leading to an increase on the number of dropped packets and the end-to-end delay. Even the links (B,C) and (C,D) are negatively affected meaning that the greedy node A is increasing its throughput in the detriment of the quality of service requirements of its own traffic flow.

On the contrary of MANET, the situation discussed above does not arise in WLAN environment since all the nodes are within the transmission range of the AP, therefore the increase of the greedy node's throughput does not affect the end-to-end delay of its traffic flow. As a conclusion, for a more effective greedy behavior the greedy node should choose an alternative strategy adapted to the constraints of MANET environment.

III. Designing new greedy strategy for MANET

In this section, we give the road map of the required steps for the greedy node to launch the greedy attack according to our strategy. First, we provide the basic assumptions of our scheme followed by a description of how the greedy node constructs the conflict graph. Next, we show how to extract the bandwidth fair share of a node according to the conflict graph. Afterwards, we determine the maximum extra bandwidth the greedy node can gain without negatively affecting its traffic flow performance. Finally, we present the algorithm used by the greedy node to launch the greedy attack and to ensure the accordance with the values computed in the previous step.

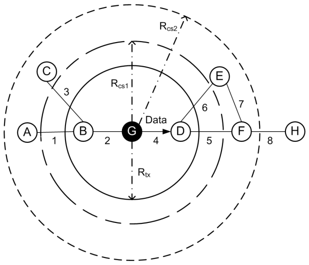


Figure 3: The connectivity graph.

A. Main Assumptions

We give an overview of the assumptions used throughout the paper. These assumptions constitute the core of our cheating strategy.

- A proactive routing protocol is used at the network layer to establish end-to-end routes such as OLSR [1][9] .
- Carrier sensing range (R_{cs}) is equal to more than twice of the transmission range (R_{tx}) [6], whereas the signal propagation follows the 2-Ray Ground Reflection Model [2] [1].
- The nodes are distributed within the topology according to the Poisson process of parameter λ [2].
- We assume CBR traffic with fixed packets size S and the transmission rate offered by the underlying MAC protocol is β. Therefore, the number of time slots (η) needed for the transmission of the packet payload is:

$$\gamma = \frac{(\frac{S}{\beta})}{\eta} \tag{1}$$

- The length of a packet P_l is defined as the number of time slots required for its successful transmission and can be expressed as follows:

$$\begin{aligned} P_l &= \frac{(NAV - T_{DATA}) + DIFS + H_l}{\eta} + \gamma \\ P_l &= \frac{Duration + DIFS + H_l}{\eta} + \gamma \end{aligned} \tag{2}$$

where

$$Duration = T_{RTS} + T_{CTS} + T_{ACK} + 3 \times SIFS$$

Notice that T_{RTS}, T_{CTS} and T_{ACK} refer to the transmission time of RTS, CTS and ACK frames respectively, whereas H_l denotes the aggregate length of Physical, MAC and UDP headers defined as follows:

$$H_l = \frac{PHS + MHS + UHS}{\beta} \tag{3}$$

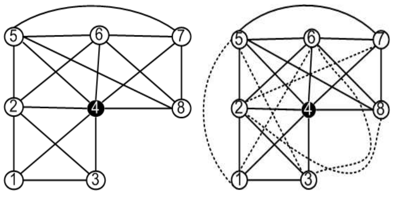
where PHS, MHS and UHS are the size of PHY, MAC and UDP headers respectively.

B. Conflict graph construction

As a first step of our scheme, the greedy node constructs the contention flow graph with nodes within its R_{cs} to derive its predicted fair-share of bandwidth [10]. To this end, the greedy node analyzes the received information in Hello and topology control (TC) messages and constructs its conflict graph [8] accordingly. For example, node G in the topology shown in Fig. 3 acquires the set of its 2-hops neighbors A, C, E and F from the Hello messages sent by nodes B

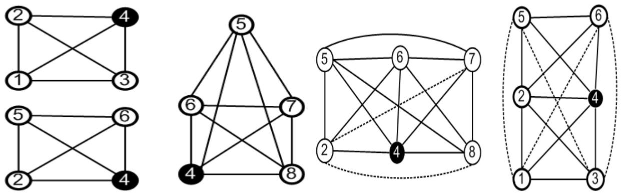
[1] Notice that we can use any other routing protocol which provides the same topology view as OLSR.

[2] The two-ray ground model is a common propagation model that has been widely used in wireless communications. Applying different propagation models could change the result, but the change is expected to be subtle.



(a) (b)

Figure 4: Conflict graph of the contending transmission. (a) R_{cs} = R_{cs1} and (b) R_{cs} = R_{cs2}.



(a) (b)

Figure 5: The set of maximal cliques. (a) R_{cs} = R_{cs1}, 3 maximal cliques whose sizes are 4, 4 and 5, respectively; (b) R_{cs} = R_{cs2}, 2 maximal cliques of 6 vertices each. Note that the dashed edges represent the new links created due to the increase of R_{cs}.

and D, and it discovers its 3-hops neighbor H from the TC message sent by the node F which is multipoint Relay (MPR) of node H.

After acquiring the necessary information, the greedy node G constructs the conflict graph within its carrier sensing range, from which it extracts the set of maximal cliques. Since the topology information acquired from Hello and TC messages is partial, node G constructs this graph by considering the worst case scenario assuming the maximum number of contending links to compute the minimum bandwidth fair share. The number of maximal cliques is the key for determining the misbehaving threshold which will be discussed later. As shown in Fig. 4, the conflict graph depends on the extent of the carrier sensing range of the greedy node, for which we distinguish two cases:

- R_{cs} is slightly larger than the transmission range R_{tx} (see Fig. 3, R_{cs} = R_{cs1}), thus we have less contention between links and consequently the greedy behavior impact reduces. According to the set of maximal cliques shown in Fig. 5(a), only a simultaneous transmission over the following pair of links is allowed: (1,5), (3,5), (2,7), (1,6), (3,6), (2,8), (1,7), (3,7), (1,8) and (3,8).
- R_{cs} is greater than twice of the transmission range, $R_{cs} > 2 \times R_{tx}$ (see Fig. 3, R_{cs} = R_{cs2}) which means that all the 2-hops neighbors of the greedy node G are within its carrier sensing range. Hence, only few links can be active for flow transmission at the same time as depicted in Fig. 5(b) where only the pairs of links $(1,7)$, $(1,8)$, $(3,7)$ and $(3,8)$ are allowed to transport traffic flows simultaneously. As compared to the first case, the number of conflict between links raises leading to devastating consequences if one node doesn't obey the MAC protocol rules.

Time complexity for generating the maximal cliques: . Given that for N nodes we have at most $\frac{N(N-1)}{2}$ links which can be established between them. According to the algorithm by Tomita et al. [13], the worst-case time complexity for generating the set of maximal cliques from the graph constructed by those links is estimated to $O(\ 3^{N/3})$.

C. Bandwidth fair-share estimation

Once the conflict graph is established and the set of maximal cliques is derived, the node G computes its fair share of bandwidth and the end-to-end throughput of its traffic flow. In order to compute these values, we assume each node has a nonempty buffer of packets ready to be transmitted at each time slot (saturation case). Hence, given a particular path relaying source and destination nodes, the end-to-end throughput capacity is defined as the minimum link throughput capacity B_i of this path. As the links in conflict with the link 4 ($G \rightarrow D$) are the bottleneck for any flow crossing them when G is cheating, we determine the end-to-end throughput E_{th} as the minimum capacity of these links. This means if the length of any path from source to destination is n hops, the end-to-end throughput is computed only for the m hops ($m \leq n$) within the sensing range of the source node, which can be formulated as

$$\begin{aligned} B_i &= sc_t \times (1-\tau)^{\lambda \pi R_{cs}^2 - 1} \times \beta \times \frac{\gamma}{P_l} \quad (4) \\ E_{th} &= min(B_1, ..., B_{\lambda \pi R_{cs}^2}) \quad (5) \end{aligned}$$

where sc_t denotes the duration of successful and collided transmissions of node i and τ is the probability of transmission. For more details on the computation of the values above, the reader may refer to the work by Wang and Garcia-Luna-Aceves [5].

Since we construct a partial topology graph limited to links in which at least one of the extremity is within the carrier sensing range of node G, then the calculated fair-share B_i of a node i might be greater than the real one R_i. That is the reason why R_i can be expressed in function of Bi as

$$R_i = B_i \times \Phi \times \Psi \quad (6)$$

such that Φ is a factor used to adjust the estimated faire-share to the real one, where

$$0.5 \leq \Phi \leq 1$$

Moreover, as the greedy node constructs the topology graph by considering the maximum number of links relaying its 2-hops neighbors and between theses nodes and its three hops neighbors, the computation of the fair-share is done based on links which might not exist. For that reason, the value Ψ, dubbed density factor, is used to increase this fair-share accordingly. This value is determined upon the following criterions:

- Nodes' density in the neighborhood of the greedy node.
- The number of MPR nodes selected by the greedy node; if this number is small and the density of nodes within its carrier sensing range is high then it is more likely to have more links between nodes, and consequently Ψ can be assigned a value close to 1. On the other hand, if the MPR set is large and the nodes' density is mediocre then the value of Ψ can be increased more than the previous case.

The value Ψ is expressed as

$$\Psi = 1 + \frac{|MPR\_set|}{|S_1 \cup S_2|} \quad (7)$$

where S_1 and S_2 denote the sets of node G's 1- and 2-hops neighbors respectively.

Remark: In our proposed strategy, one may argue that a node can request its one or 2-hops neighbors for exchanging topology information in order to get the complete view of the network. However, such an action makes the node suspicious which may facilitate its detection if any anti MAC layer misbehavior system is deployed. Moreover, none of its neighbors will respond to these requests since they are not considered proper operations of the routing protocols. As a consequence, our proposed algorithm is more secure and realistic since it depends only on the information gathered locally by the greedy node from the legitimate exchange of control packets.

D. Misbehaving threshold computation

In this section, we define an upper bound of the extra bandwidth earned by the greedy node, dubbed misbehaving threshold. Any greedy node overtaking this threshold will experience a decrease of its own flows' performance (in terms of end-to-end throughput and delay).

As known, in the case where fair-share is held amongst the contending nodes, the greedy node gets R_i of bandwidth. When the greedy node misbehaves, its share is R_i + B_g which means that it acquires B_g of extra bandwidth share as a result of its mischief. So, for a rational greedy node, the value B_g should satisfy the following condition

$$\begin{aligned} \frac{[B-(R_i+B_g)]}{(N-1)} &> \quad \alpha \times E_{th} \\ (B-R_i-B_g) &> \quad (N-1) \;\; \alpha \times E_{th} \end{aligned}$$

therefore

$$B_g \leq B - R_i - (N-1) \;\; \alpha \times E_{th} \tag{8}$$

such that, N is the average number of nodes within the carrier sensing range, R_{cs}, of the greedy node, which can be expressed as $N = \lambda \pi R_{cs}^2$. B is the total bandwidth available and E_{th} is the estimated end-to-end throughput of the ongoing flow calculated according to the formula given in Eq. 5.

The reason of using the condition above is the fact that any adopted misbehaving strategy which reduces the mean [3] of the greedy node neighbors' throughput below the value of E_{th} has also a negative impact on its own flow's performance. Hence, the rational greedy node has to ensure that B_g fulfils the condition given in Eq. 8 in order to satisfy the QoS requirements of its flow.

Notice that the value α is used to adjust the extra bandwidth gain of the greedy node with respect to the topology of the bottleneck area (the area covered by R_{cs}) and the contention flow graph. It can be expressed as

$$\alpha = \frac{\sum_{i=1}^{N} MC_i}{N \times cl} \tag{9}$$

where MC_i denotes the number of maximal cliques to whom the link $i \leftrightarrow j$ belong such that the node i is either sender or receiver over this link, and cl is the total number of the maximal cliques in the conflict graph.

E. Launching the rational greedy strategy

Once all the parameters defined in the previous steps are computed, the greedy node G carries out the two misbehavior techniques described below to achieve its goal.

It first selects a small Backoff value in order to gain more bandwidth within its allowed threshold. Then, it provokes collisions with its neighbors' frames except the frames of its ongoing flow's next hop by simply scheduling a transmission of a small or empty packet whenever it receives an RTS which is not destined to it and not sent by its next hop node. This process is illustrated by the Algorithm 1.

These two steps needs to be adjusted according to the misbehaving threshold, B_g, which means that the greedy node must compute the bandwidth share acquired by its next hop and adjusts its jamming rate and contention window accordingly. This process is described in Algorithm 2. In this algorithm, the estimation of the next hop's bandwidth is periodically computed whenever the timer period is expired.

Algorithm 1 Greedy node behavior

```
if ($RTS$ received) then
    $attempt$ = $false$;
    if (@Dest == my address) then
        schedule $CTS$ transmission;
    else
        if ( @source != @NH ) then
            if ($++CPT < n_1$) then
                schedule transmission of empty or small packet
                after SIFS;
            end
            $CPT$ = $CPT$ $mod$ $n_2$;
        else
            $attempt$ = $true$;
        end
    end
end
```

/*where @NH is the greedy node's next hop for the ongoing flow. n_1, n_2 and CPT are values used to adjust the jamming rate such that $n_1 < n_2$. */

Algorithm 2 Next hop bandwidth estimation and adjustment of the cheating parameters

```
if (($DATA$ received) && ($attempt$ == $true$) then
    $B_{NH}$ = $B_{NH}$ + $P_l$;
    if (Period elapsed) then
        if ($B_{NH} < E_{th}$ ) then
            increase jam rate;
            if ($B_{own} > R_{share}$ +$B_g$) then
                decrease $k$;
            end
        else
            if (($E_{th}$-$B_{NH}$) > $threshold$) then
                decrease jam rate;
            end
        end
        $B_{NH}$ = 0;
        $B_{own}$ = 0;
    end
end
```

/*B_{NH} and B_{own} are the bandwidth gained by the next hop and the greedy node respectively, expressed in number of time slots, during each period. The value $Threshold$ is used by the greedy node to prevent wasting more energy in jamming whenever its goal is achieved. k is the misbehavior coefficient used to choose a small backoff value, for more details you may refer to our previous work in [16]. */

IV. Experimental study

We now proceed to the experimental evaluation of our proposed greedy strategy. First, we illustrate the propagation of the greedy behavior's impact in ad hoc networks. Then, we emphasize the benefits gained by the greedy node in terms of throughput, end-to-end delay, and delivery ratio of its traffic flow when it behaves according to our strategy. The simulation parameters are summarized in Table I.

A. Propagation of Greedy Behavior Impact

In our experiments, we consider the same topology shown in Fig. 3 where two traffic flows are generated, G → A and F → H. The traffic sources send 1000 bytes every 2 ms (500 packets/s each) which

[3] We use the mean of throughput of the greedy node's neighbors as there is a common bandwidth fair-share for each of them.

Parameters	Values
Area	2000m × 1000m
Physical layer	direct sequence
Transmission range	250m
Carrier sensing range	550 m
Traffic type	CBR
Data rate	2 mbps
CBR packets size	500 bytes
Buffer size	64 packets
Simulation time	300 seconds
# simulation epochs	5
Network simulator	OPNET 14.0 [17]

Table I: Simulation settings

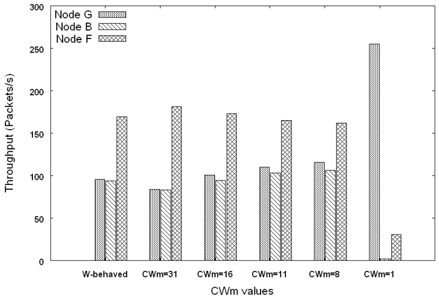


Figure 6: Propagation of greedy behavior's impact according to CW_m variation in MANET, measured in terms of the acquired throughput.

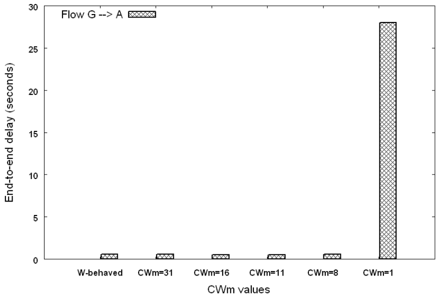


Figure 7: End-to-end delay of the greedy node's flow versus CW_m size.

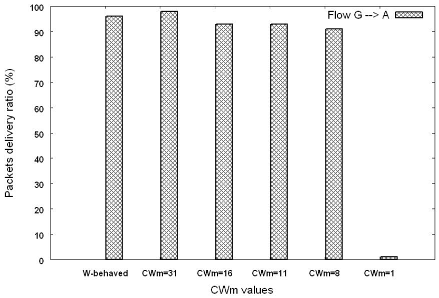


Figure 8: Variation of the packet delivery ratio of the greedy node's flow versus the chosen CW_m value

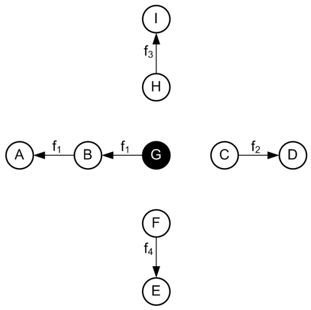


Figure 9: Topology used for evaluation of our proposed greedy behavior strategy.

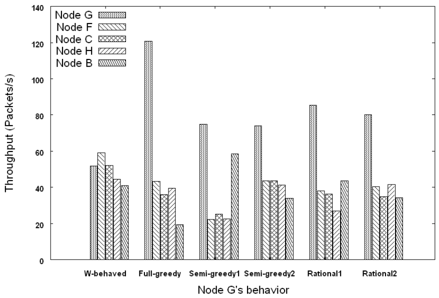


Figure 10: Variation of the traffic flows sources' throughput with the different cheating strategies adopted by the greedy node G.

means each source node has a packet ready for transmission at each time slot. Fig. 6 plots the obtained throughput by the greedy node G, its next hop node B and the node F with different values of the contention window of node G. When node G behaves correctly or sets its contention window constantly to 31 (equivalent to the minimum contention window CW_m), node F gets more bandwidth since it has less contention than node G. As we can see from the network topology, the location of node F favors it to seize the channel more likely than nodes G and B, leading to short term unfairness as well as long term unfairness. For example, during node B's transmission node F monopolizes the channel by transmitting continuously over the link 8 (all links in conflict with this link are inactive) and then increases its chance to transmit before node G which is deferring due to node B's transmission.

The throughput earned by the node F decreases slightly with the decrease of node G's contention window, whereas the throughput of nodes G and B is increasing, until it collapses sharply when node G's contention window is set to 1. When node G monopolizes the medium by choosing constantly a backoff value equal to 0, CW_m = 1, the throughput of its next hop, node B, drops sharply and consequently the delivery ratio of the flow G → A drops as well (see Fig. 8.)

From Figures 6, 7 and 8 we note that the misbehavior of the node G has a devastating impact only when it constantly sets its CW_m to 1 where the end-to-end delay for the small portion of packets forwarded by node B becomes quite long leading to the violation of the running application's QoS requirements. Moreover, this impact propagates to affect any other traffic flow within nodes G's carrier sensing range which makes this area a bottleneck in the network.

B. Advantages of the proposed greedy behavior strategy

In this section, we highlight the advantage of adopting our strategy by the greedy node. We consider the topology shown in Fig. 9, where four traffic flows f_1, f_2, f_3 and f_4 are generated in the

	W-behaved	Full-greedy	Semi-greedy1	Semi-greedy2	Rational1	Rational2
		CW_m = 1	CW_m = 31, jam_rate = 0.5	CW_m = 16, jam_rate = 0.2		
End-to-end delay (seconds)	0.679	1.4295	0.554	1.1388	0.859	0.985
Delivery ratio (%)	79.11	15.95	76.59	46.04	50.59	45.93

Table II: End-to-end delay and packet delivery ratio of flow f_1 under various greedy behavior strategies.

network such that each source node sends 200 packets per second of 500 bytes each. In this scenario, we vary the node G's behavior among different strategies and observe the impact of each on the throughput, end-to-end delay and the packet delivery ratio. From the simulation results obtained in Fig. 10, we can see that when the node G tries to monopolize the medium for its own traffic (Full-greedy), its throughput gain is more than twice of the one earned in the W-behaved case and the bandwidth gained by its next hop node B is decreased to less than half. Consequently, the end-to-end delay of the flow f_1 is doubled along with the collapse of the packet delivery ratio as depicted in Table II. Hence, as opposed to WLAN the Full-greedy strategy is inadequate in MANET since it affects the performance of the traffic flow initiated by the greedy node itself.

As alternative strategies, we have implemented Semi-greedy1 and Semi-greedy2 in which the node G constantly sets its CW_m to 31 and 16, its jam-rate to 0.5 and 0.2, respectively. The results show that in the former strategy node G successfully increases its own throughput and the one of its next hop compared to the W-behaved strategy whereas the throughput of all its neighbors decreases to less than half. The drawback of this strategy is the energy necessary to jam 50 % of CTS packets sent by its 2-hops neighbors. Hence, due to the limited energy in MANET, this strategy is unsuitable for adoption by the node G. For the latter strategy, node B's throughput is increased considerably along with the delivery ratio compared to the Full-greedy strategy; however, the rapidly changing topology of MANET makes it inefficient since the chosen jam-rate and CW_m may not produce the same results in different network topologies.

Based on our discussion above, the main issues for choosing a suitable greedy strategy in MANET are the energy constraints and the rapidly changing topology. To circumvent the limitations of the previous strategies regarding these issues, we apply our proposed method where we have implemented two scenarios Rational1 and Rational2. In Rational1, the greedy node G constructs the conflict graph according to the information acquired from HELLO and TC messages (i.e., the best case in terms of the obtained throughput), whereas in Rational2 it assumes the maximum number of contention among the links (i.e., the worst case for the estimated throughput). As shown in Fig. 10 and Table II, both scenarios give good results in terms of end-to-end delay and packet delivery ratio as compared to Full-greedy strategy, with a considerable increase of throughput where node B's throughput is almost equal to the one acquired in W-behaved strategy. Moreover, in both scenarios the greedy node G still gains more bandwidth than its neighbors and maintains a reasonable performance of its flow f_1. Therefore, these results prove the efficiency of our greedy strategy in MANET.

V. Conclusion

In this work, we have analyzed the greedy behavior problem in wireless ad hoc networks and proved that its impact can be more devastating compared to wireless local area networks. The propagation of the effect of this misbehavior is illustrated through conflict graphs analysis. As a result of this investigation, an effective greedy behavior strategy is proposed suitable for ad hoc networks. This method allows the greedy node to gain more bandwidth share compared to its neighbors and keeps the performance of its ongoing flows reasonable by maintaining its extra bandwidth share within the misbehaving threshold. Our algorithm is evaluated through extensive simulations and the obtained results highlight its advantage in terms of the increase in delivery ratio and the reduction of the end-to-end delays compared to the Full-greedy strategy applied in WLAN.

References

[1] W.Y. Lee, "Mobile Communications Engineering", McGraw-Hill 1982, ISBN 0-07-037039-7.

[2] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals", *IEEE Transactions on Communications*, 32(3):246-257, March 1984.

[3] "IEEE 802.11 wireless LAN media access control (MAC) and physical layer (PHY) specifications", ANSI/IEEE Std 802.11, 1999.

[4] V. Gupta, S. Krishnamurthy and M. Faloutsos, "Denial of service attacks at the MAC layer in wireless ad hoc networks", *In Proc. of Military Communication Conference (MILCOM'02)*, October 2002, pp. 1118-1123.

[5] Y. Wang and J.J. Garcia-Luna-Aceves, "Performance of Collision Avoidance Protocols in Single-Channel Ad Hoc Networks", *In Proc. of the* 10^{th} *IEEE International Conference on Network Protocols (ICNP'02)*, November 2002, pp. 68-77.

[6] K. Xu, M. Gerta and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks", *Ad Hoc Networks Journal*, 1(1):107-123, July 2003.

[7] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions", *In Proc. of the* 12^{th} *USENIX Security Symposium*, August 2003.

[8] K. Jain, J. Padhye, V. Padmanabhan and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance", *ACM/Springer Journal of Wireless Networks*, 11(4):471-484, July 2005.

[9] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", *IETF RFC 3626 (Experimental)*, October 2003.

[10] Z. Fang and B. Bensaou, "Fair bandwidth sharing algorithms based on game theory frameworksfor wireless ad-hoc networks", *In Proc. of IEEE INFOCOM'04*, March 2004, pp. 1284-1295.

[11] A. A. Cardenas, S. Radosavac and J. S. Baras, "Detection and Prevention of MAC layer misbehavior in Ad Hoc Networks", *In Proc. of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, October 2004, pp. 17-22.

[12] P. Kyasanur, and N. H. Vaidya, "Selfish MAC Layer misbehavior in Wireless Networks", *IEEE Transactions on Mobile Computing*, 4(5):502-516, September-October 2005.

[13] E. Tomita, A. Tanaka and H. Takahashi, "The Worst-Case Time Complexity for Generating All Maximal Cliques", *Theoretical Computer Science Journal of Elsevier*, 363(1):28-42, October 2006.

[14] M. Raya, I. Aad, J. P. Hubaux and A. El Fawal, "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots", *IEEE Transactions on Mobile Computing*, 5(12):1691-1705, December 2006.

[15] L. Guang, C. M. Assi and A. Benslimane, "Enhancing IEEE 802.11 Random Backoff in Selfish Environments", *IEEE Transactions on Vehicular Technology*, 57(3):1806-1822, May 2008.

[16] S. Djahel and F. Nait-Abdesselam, "A Fuzzy Logic Based Scheme to Detect Adaptive Cheaters in Wireless LAN", *In Proc. of International Conference on Communications (ICC'09)*, June 2009.

[17] OPNET Technologies. *OPNET Modeler.* http://www.opnet.com/.